

Proyecto final MP

El proyecto ha sido realizado en solitario por Juan Manuel Rubio Rodríguez, perteneciente al grupo de prácticas del viernes.

Desarrollo de la práctica

La práctica en sí misma no ha representado grandes problemas. En general considero que se ha aportado un diseño correcto y en consonancia con los objetivos del proyecto. La implementación se ha desarrollado de forma paulatina haciendo un seguimiento del material ofrecido por los docentes, sin apenas contratiempos, más allá de errores puntuales de programación (no conceptuales).

Por otra parte, y de cara a la ejecución del proyecto, cabe resaltar que sí he sufrido una pérdida considerable de tiempo al utilizar la herramienta Netbeans, la cual no me parece que aporte una mejora significativa con respecto a cualquier editor de texto plano para desarrollar un proyecto de las dimensiones de éste.

Quiero resaltar que durante el desarrollo, he sufrido lo que voy a catalogar de “bug” debido a la incapacidad para descubrir el error subyacente (si lo hubiere), el cual dejó de afectar al proyecto sin tener que cambiar una línea de código.

Respecto al código, dos aspectos principales:

- No he comprobado, al utilizar memoria dinámica, si el puntero era distinto de cero. Actualmente el lenguaje soporta realizar un delete sobre un puntero inicializado a cero.
- La solución por la que he optado para implementar constructor de copia y operador de asignación de la clase ConjuntoParticulas para no repetir código, ha sido la de implementar el operador de asignación haciendo uso del constructor de copia y posteriormente realizar un swap.

Opciones adicionales

Se han añadido las opciones que se sugerían en el guión en relación al aumento/disminución de partículas dinámico.

La opción 3 muestra el simulador original sin modificaciones en su comportamiento.

La opción 4 muestra un simulador modificado de acuerdo a las especificaciones comentadas al final de este epígrafe.

Se han implementado sobre la opción 3 y 4 (véase main.cpp).

1. En caso de pulsar la tecla ESCAPE: se regenera el simulador, volviendo a crear los conjuntos de partículas de forma aleatoria.
2. En caso de pulsar F1, se agrega una partícula fija al conjunto.
3. En caso de pulsar F2, se agrega una partícula móvil al conjunto.

Se ha cambiado el método GestionarColisiones de la clase ConjuntoParticulas para que en caso de colisionar dos partículas del mismo color, una de ellas desaparezca.

Se ha modificado el método Rebotes de la clase simulador para que, en caso de colisionar una partícula móvil con una fija, si son del mismo color, se genere otra partícula móvil (del mismo color, pero en una posición aleatoria) y, en caso contrario, que desaparezca la partícula móvil.

Valgrind

Opción 1:

```

Juanma@Juanma-VivoBook-ASUSLaptop-X430FA-S430FA: ~/NetBeansProjects/P6/dist/Debug/GNU-Linux
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
==19731==
Opcion a probar: 1

** Prueba sobrecarga de operadores **
posX: 25 posY: 283 color: 7      posX: 483 posY: 496 color: 3      posX: 149 posY: 418 color: 1      posX: 558 posY: 130 color: 8
posX: 235 posY: 290 color: 6      posX: 254 posY: 309 color: 6
posX: 235 posY: 290 color: 6      posX: 254 posY: 309 color: 6      posX: 46 posY: 304 color: 7      posX: 487 posY: 500 color: 3      posX: 161 posY: 430 color: 1      posX: 577 posY: 149 color: 8
posX: 235 posY: 290 color: 6      posX: 254 posY: 309 color: 6      posX: 46 posY: 304 color: 7      posX: 487 posY: 500 color: 3      posX: 161 posY: 430 color: 1      posX: 577 posY: 149 color: 8      posX: 46 po
color: 7      posX: 487 posY: 500 color: 3      posX: 161 posY: 430 color: 1      posX: 577 posY: 149 color: 8

^C==19731==
==19731== Process terminating with default action of signal 2 (SIGINT)
==19731== at 0x54C4100: __lll_lock_wait (lowlevellocks.5:135)
==19731== by 0x5ABD022: pthread_mutex_lock (pthread_mutex_lock.c:78)
==19731== by 0x60CEE77: xcb_writev (in /usr/lib/x86_64-linux-gnu/libxcb.so.1.1.0)
==19731== by 0x4E7A145: _XSend (in /usr/lib/x86_64-linux-gnu/libX11.so.6.3.0)
==19731== by 0x4E7A66B: _XReply (in /usr/lib/x86_64-linux-gnu/libX11.so.6.3.0)
==19731== by 0x4E761CC: XSync (in /usr/lib/x86_64-linux-gnu/libX11.so.6.3.0)
==19731== by 0x4E56E1D: XCloseDisplay (in /usr/lib/x86_64-linux-gnu/libX11.so.6.3.0)
==19731== by 0x108E69: main (minlwin.cpp:740)
==19731==
==19731== HEAP SUMMARY:
==19731==    in use at exit: 53,642 bytes in 34 blocks
==19731==   total heap usage: 373 allocs, 339 frees, 142,852 bytes allocated
==19731==
==19731== 288 bytes in 1 blocks are possibly lost in loss record 22 of 31
==19731== at 0x4C31B25: calloc (in /usr/lib/valgrind/vgpreload_memcheck-and64-linux.so)
==19731== by 0x40134A6: allocate_dtv (dl-tls.c:286)
==19731== by 0x40134A6: _dl_allocate_tls (dl-tls.c:530)
==19731== by 0x5AB8227: allocate_stack (allocatestack.c:627)
==19731== by 0x5AB8227: pthread_create@@GLIBC_2.2.5 (pthread_create.c:644)
==19731== by 0x108C80: _maybe_call_main() (minlwin.cpp:673)
==19731== by 0x108D02F: _process_event() (minlwin.cpp:689)
==19731== by 0x108E21: main (minlwin.cpp:733)
==19731==
==19731== LEAK SUMMARY:
==19731==    definitely lost: 0 bytes in 0 blocks
==19731==    indirectly lost: 0 bytes in 0 blocks
==19731==    possibly lost: 288 bytes in 1 blocks
==19731==    still reachable: 53,354 bytes in 33 blocks
==19731==    suppressed: 0 bytes in 0 blocks
==19731== Reachable blocks (those to which a pointer was found) are not shown.
==19731== To see them, rerun with: --leak-check=full --show-leak-kinds=all
==19731==
==19731== For counts of detected and suppressed errors, rerun with: -v
Juanma@Juanma-VivoBook-ASUSLaptop-X430FA-S430FA: ~/NetBeansProjects/P6/dist/Debug/GNU-Linux$
```

Opción 2:

```
juanma@juanma-VivoBook-ASUSLaptop-X430FA-S430FA: ~/NetBeansProjects/P6/dist/Debug/GNU-Linux -
Archivo Editar Ver Buscar Terminal Ayuda
juanma@juanma-VivoBook-ASUSLaptop-X430FA-S430FA:~/NetBeansProjects/P6/dist/Debug/GNU-Linux$
valgrind --leak-check=full --track-origins=yes ./p6
==20207== Memcheck, a memory error detector
==20207== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==20207== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==20207== Command: ./p6
==20207==

Opcion a probar: 2

** Prueba Conjunto Particulas **
==20207==
==20207== HEAP SUMMARY:
==20207==    in use at exit: 448 bytes in 2 blocks
==20207==    total heap usage: 1,586 allocs, 1,584 frees, 474,716 bytes allocated
==20207==
==20207== 288 bytes in 1 blocks are possibly lost in loss record 2 of 2
==20207==    at 0x4C31B25: calloc (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so)
==20207==    by 0x40134A6: allocate_dtv (dl-tls.c:286)
==20207==    by 0x40134A6: _dl_allocate_tls (dl-tls.c:530)
==20207==    by 0x5ABB227: allocate_stack (allocatestack.c:627)
==20207==    by 0x5ABB227: pthread_create@@GLIBC_2.2.5 (pthread_create.c:644)
==20207==    by 0x10BCBD: _maybe_call_main() (miniwin.cpp:673)
==20207==    by 0x10BD2F: _process_event() (miniwin.cpp:689)
==20207==    by 0x10BE21: main (miniwin.cpp:733)
==20207==
==20207== LEAK SUMMARY:
==20207==    definitely lost: 0 bytes in 0 blocks
==20207==    indirectly lost: 0 bytes in 0 blocks
==20207==    possibly lost: 288 bytes in 1 blocks
==20207==    still reachable: 160 bytes in 1 blocks
==20207==    suppressed: 0 bytes in 0 blocks
==20207== Reachable blocks (those to which a pointer was found) are not shown.
==20207== To see them, rerun with: --leak-check=full --show-leak-kinds=all
==20207==
==20207== For counts of detected and suppressed errors, rerun with: -v
==20207== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 200 from 1)
juanma@juanma-VivoBook-ASUSLaptop-X430FA-S430FA:~/NetBeansProjects/P6/dist/Debug/GNU-Linux$
```

Opción 3:

```
juanma@juanma-VivoBook-ASUSLaptop-X430FA-S430FA: ~/NetBeansProjects/P6/dist/Debug/GNU-Linux -
Archivo Editar Ver Buscar Terminal Ayuda
juanma@juanma-VivoBook-ASUSLaptop-X430FA-S430FA:~/NetBeansProjects/P6/dist/Debug/GNU-Linux$
valgrind --leak-check=full --track-origins=yes ./p6
==20219== Memcheck, a memory error detector
==20219== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==20219== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==20219== Command: ./p6
==20219==

Opcion a probar: 3

***** Probando Simulador *****
==20219==
==20219== HEAP SUMMARY:
==20219==   in use at exit: 448 bytes in 2 blocks
==20219==   total heap usage: 2,165 allocs, 2,163 frees, 191,400 bytes allocated
==20219==
==20219== 288 bytes in 1 blocks are possibly lost in loss record 2 of 2
==20219==   at 0x4C31B25: calloc (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so)
==20219==   by 0x40134A6: allocate_dtv (dl-tls.c:286)
==20219==   by 0x40134A6: _dl_allocate_tls (dl-tls.c:530)
==20219==   by 0x5ABB227: allocate_stack (allocatestack.c:627)
==20219==   by 0x5ABB227: pthread_create@@GLIBC_2.2.5 (pthread_create.c:644)
==20219==   by 0x10BCBD: _maybe_call_main() (miniwin.cpp:673)
==20219==   by 0x10BD2F: _process_event() (miniwin.cpp:689)
==20219==   by 0x10BE21: main (miniwin.cpp:733)
==20219==
==20219== LEAK SUMMARY:
==20219==   definitely lost: 0 bytes in 0 blocks
==20219==   indirectly lost: 0 bytes in 0 blocks
==20219==   possibly lost: 288 bytes in 1 blocks
==20219==   still reachable: 160 bytes in 1 blocks
==20219==   suppressed: 0 bytes in 0 blocks
==20219== Reachable blocks (those to which a pointer was found) are not shown.
==20219== To see them, rerun with: --leak-check=full --show-leak-kinds=all
==20219==
==20219== For counts of detected and suppressed errors, rerun with: -v
==20219== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 500 from 1)
juanma@juanma-VivoBook-ASUSLaptop-X430FA-S430FA:~/NetBeansProjects/P6/dist/Debug/GNU-Linux$
```